Project - Network Simulator

Swastik Gupta(BITE013),Manish Kumar(BITE043),Gurveen Singh(BITE005)

0.1 Objective:

Demonstrating the working of all the layers of Network Protocol Stack using Network Simulator

0.2 Programming Language used:

C++

0.3 Details:

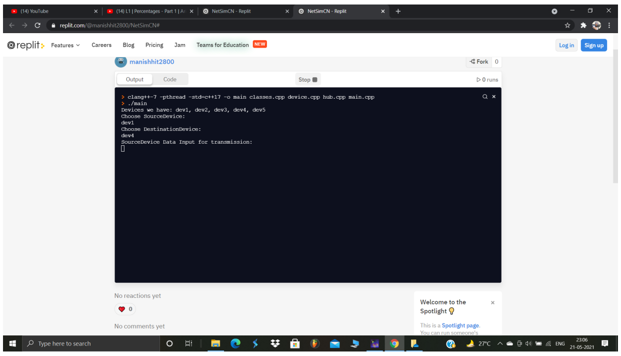
- Object oriented programming concepts like creation of classes and objects is used.

  - Class **Device** - MACadress , data , setMACaddress and seddata are the attributes.The objects created of this class are the nodes in the network.

  - Class **Hub** - data2, macAdd ,store and hub2dev are the attributes.The object created (hb1) keeps a check on whether data is revieved at the correct device or not from the sender.

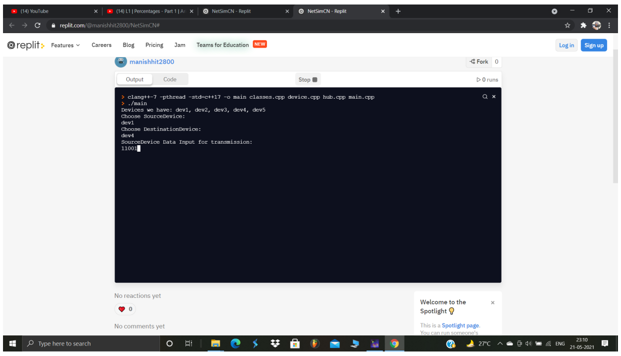
- The simulation in this project shows the flow of data from one node to other.

- Firstly, the user is expected to choose the source device and the destination device.



```
> clang++-7 -pthread -std=c++17 -o main classes.cpp device.cpp hub.cpp main.cpp
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
```

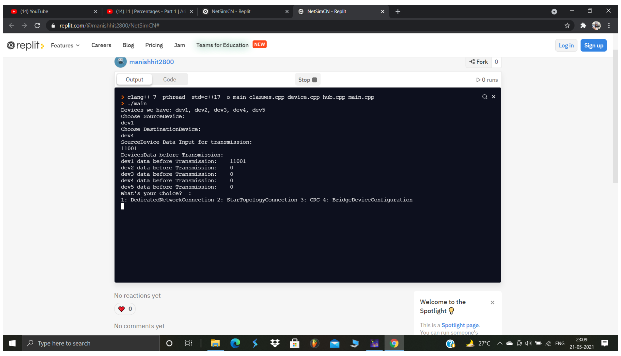
Figure 1: User chooses the source and destination device



```
> clang++-7 -pthread -std=c++17 -o main classes.cpp device.cpp hub.cpp main.cpp
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
11001
```

Figure 2: User is expected to enter the input

- Once this is done , the user is asked for data he/she wants to tranfer. The data is tranfered in the form of bits.



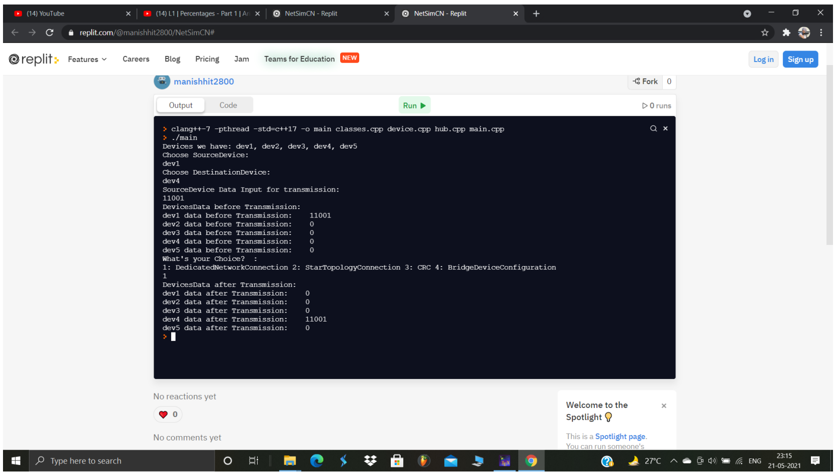
```
> clang++-7 -pthread -std=c++17 -o main classes.cpp device.cpp hub.cpp main.cpp
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
11001
DevicesData before Transmission:
dev1 data before Transmission:    11001
dev2 data before Transmission:    0
dev3 data before Transmission:    0
dev4 data before Transmission:    0
dev5 data before Transmission:    0
What's your Choice?  :
1: DedicatedNetworkConnection 2: StarTopologyConnection 3: CRC 4: BridgeDeviceConfiguration
```

Figure 3: Data each device contains before the transmission

0.4 Choice for the user:

In physical layer, transmission of bits from is from one device to other. The user is given two options for the transmission of data.They are -

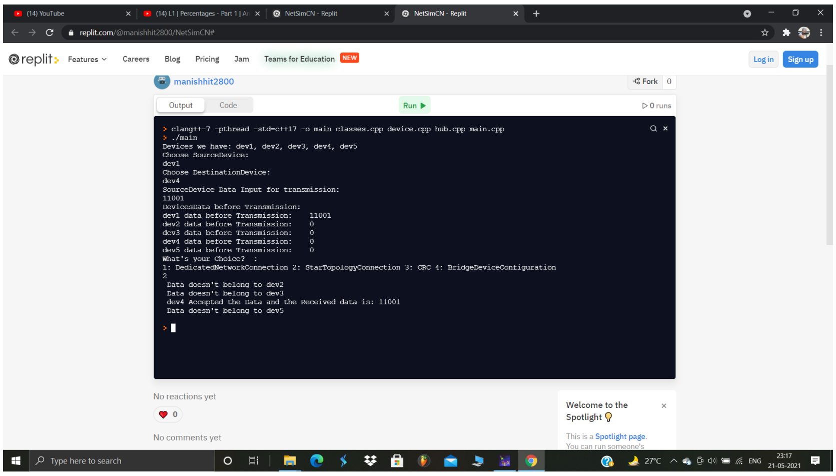
- **Using a Dedicated network connection**



```
> clang++-7 -pthread -std=c++17 -o main classes.cpp device.cpp hub.cpp main.cpp
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
11001
DevicesData before Transmission:
dev1 data before Transmission:    11001
dev2 data before Transmission:    0
dev3 data before Transmission:    0
dev4 data before Transmission:    0
dev5 data before Transmission:    0
What's your Choice?  :
1: DedicatedNetworkConnection 2: StarTopologyConnection 3: CRC 4: BridgeDeviceConfiguration
1
DevicesData after Transmission:
dev1 data after Transmission:    0
dev2 data after Transmission:    0
dev3 data after Transmission:    0
dev4 data after Transmission:    11001
dev5 data after Transmission:    0
>
```

Figure 4: Dedicated network

- **Using a Star topology connection**

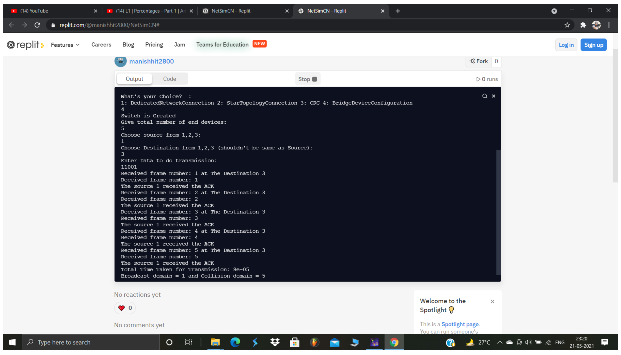


```
> clang++-7 -pthread -std=c++17 -o main classes.cpp device.cpp hub.cpp main.cpp
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
11001
DevicesData before Transmission:
dev1 data before Transmission:    11001
dev2 data before Transmission:    0
dev3 data before Transmission:    0
dev4 data before Transmission:    0
dev5 data before Transmission:    0
What's your Choice?  :
1: DedicatedNetworkConnection 2: StarTopologyConnection 3: CRC 4: BridgeDeviceConfiguration
2
 Data doesn't belong to dev2
 Data doesn't belong to dev3
 dev4 Accepted the Data and the Received data is: 11001
 Data doesn't belong to dev5

>
```

Figure 5: Star topology

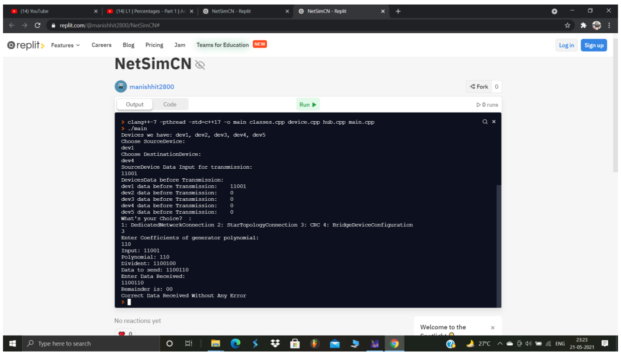
- **Bridge device configuration** - This is functionality of Data link layer.In this configuration , switch is being created and the user is asked to enter the total number of end devices.Also once the reciever recieves the frame, it send back an ACK to the source.
  Total time for transmission is shown after the transmission is complete. The user is then asked if he/she wants to more transmission.



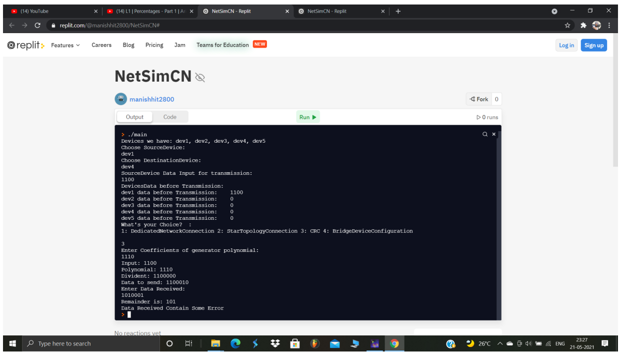
```
What's your Choice?  :
1: DedicatedNetworkConnection 2: StarTopologyConnection 3: CRC 4: BridgeDeviceConfiguration
4
Switch is Created
Give total number of end devices:
5
Choose source from 1,2,3:
1
Choose Destination from 1,2,3 (shouldn't be same as Source):
3
Enter Data to do transmission:
11001
Received frame number: 1 at The Destination 3
Received frame number: 1
The source 1 received the ACK
Received frame number: 2 at The Destination 3
Received frame number: 2
The source 1 received the ACK
Received frame number: 3 at The Destination 3
Received frame number: 3
The source 1 received the ACK
Received frame number: 4 at The Destination 3
Received frame number: 4
The source 1 received the ACK
Received frame number: 5 at The Destination 3
Received frame number: 5
The source 1 received the ACK
Total Time Taken for Transmission: 8e-05
Broadcast domain = 1 and Collision domain = 5
```

Figure 6: Functioning of Bridge device configuration

- **Error Control** - Achieved by using the concept of **CRC** .
  The user is asked to enter the source input and the coefficients of the generator polynomial.
  After this, the data to be sent is calculated.
  The user is then expected to enter the recieved data and therefore the remainder is calculated.
  If the remainder is in the form of string of 0's, there is no error.
  If there is atleast one 1 , there is a error.
  We have shown below two test cases for error detection.



```
> clang++-7 -pthread -std=c++17 -o main classes.cpp device.cpp hub.cpp main.cpp
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
11001
DevicesData before Transmission:
dev1 data before Transmission:    11001
dev2 data before Transmission:    0
dev3 data before Transmission:    0
dev4 data before Transmission:    0
dev5 data before Transmission:    0
What's your Choice?  :
1: DedicatedNetworkConnection 2: StarTopologyConnection 3: CRC 4: BridgeDeviceConfiguration
3
Enter Coefficients of generator polynomial:
110
Input: 11001
Polynomial: 110
Divident: 1100100
Data to send: 1100110
Enter Data Received:
1100110
Remainder is: 00
Correct Data Received Without Any Error
> 
```



```
> ./main
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
1100
DevicesData before Transmission:
dev1 data before Transmission:    1100
dev2 data before Transmission:    0
dev3 data before Transmission:    0
dev4 data before Transmission:    0
dev5 data before Transmission:    0
What's your Choice?  :
1: DedicatedNetworkConnection 2: StarTopologyConnection 3: CRC 4: BridgeDeviceConfiguration

3
Enter Coefficients of generator polynomial:
1110
Input: 1100
Polynomial: 1110
Divident: 1100000
Data to send: 1100010
Enter Data Received:
1010001
Remainder is: 101
Data Received Contain Some Error
> 
```

Figure 7: Test cases for error detection